Reconfigurable Packet Scheduling for Radio Access Jointly Adaptive to Traffic and Channel

A. Baiocchi, F. Cuomo, T. Melodia, A. Todini, F. Vacirca

University of Roma "La Sapienza", INFOCOM Dept., Via Eudossiana 18, 00184, Rome, Italy

Abstract. Adaptive packet scheduling for wireless data access systems, including channel state information based algorithms, is a quite well established topic. We develop here an original framework to define such an algorithm in a reconfigurable context, i.e. with the ability to exploit heterogeneous communication environments. The focus is to define platform independent algorithms that can be "adapted", by a specific software, to different environments, so that the core communication functions can be defined, modified and improved once for all. The specific communication function addressed in this work is packet scheduling. For better performance the packet scheduling design exploits the cross-layering approach, i.e. information and functions from different communication layers are used. The concept is proved with reference to the UMTS and Bluetooth technologies, as representatives of a cellular system and a local access one respectively.

1 Introduction

The importance of reconfigurability in wireless access interfaces (at least at the physical layer, e.g. software radio) is due to existence of various technologies spanning from personal area networks and wireless local area networks (e.g. IEEE 802.11, Bluetooth, Hiperlan/2) to cellular systems (e.g. GSM, GPRS, CDMA2000, UMTS, etc.).

The common meaning of reconfigurability [1] is the capability to provide a network element with different sets of communication functions and with the ability to switch among them so as to exploit any of a number of different wireless access environments that might be available at a given time and place. In the following we adopt a somewhat broader point of view: a system is reconfigurable if some communication functions and relevant algorithms are defined and implemented by abstracting from a specific wireless access. The specific wireless access is represented by models able to express the traffic load, the channel conditions and other communication parameters from a high level point of view. It becomes possible to design algorithms (e.g. packet scheduling) according to some general criteria independent of the specific technology.

A strictly related approach to high performance wireless access system is cross-layering [2] [3]. Optimization of data transfer and quality of service is best achieved by jointly considering functions and parameters traditionally belonging to different architectural layers (e.g. error control by means of hybrid ARQ/FEC techniques, power control, packet scheduling and radio resource sharing).

The conception of communication algorithms and specifically of radio resource management algorithms is therefore "generalized" along two directions: reconfigurability, in order to exploit different wireless communication systems, and cross-layer design, to achieve a better efficiency and exploitation of the time-varying communication environment.

This work fits in a comprehensive effort to specify a reconfigurable wireless communication architecture for mobile computing applications, named *Re-configurable Access module for MObile computiNg applications* (RAMON in the sequel [1]). The framework of RAMON is a mobile computing environment adopting the TCP/IP protocol suite. Different communication environments (*Reference Environment*, RE) are provided to the mobile host; in the following, we consider as example network environments UMTS-TDD [4] and Bluetooth [5].

This paper focuses on the radio resource management functions, specifically on packet scheduling and resource sharing. There are some works on these subjects with specific reference to the wireless access. Papers [6]-[8] address wireless packet scheduling with emphasis on the fairness issue arising primarily from reuse of channel state information and the short term variability of channel condition that bring about unfairness at least in the short term if a high efficiency is targeted. Adaptation of the classic Weighted Fair Queuing schemes are investigated in [6] and [7]. Papers [9]-[12] deal with radio resource sharing for wireless packet access in a CDMA based air interface with mutual interference. Finally, papers [13] and [14] consider using channel state information for resource sharing in wireless packet access.

Some of these works take what can be viewed as a cross-layer approach, in that radio resource assignment is based on radio channel state for the competing links. This is also the approach considered in our research [15].

The novel points with respect to previous works are the reconfigurability framework and the specific scheduling algorithm with its adaptation to the considered REs (UMTS and Bluetooth). The first point has a conceptual value: from a large number of studies developed with reference to specific technologies, we abstract the major common ideas to define a general framework for wireless access thus simplify the design of efficient algorithms. The second point is the development of a packet scheduling scheme that is jointly adaptive to channel and traffic conditions. This scheme, named *CHannel Adaptive Open Scheduling* (CHAOS) fully exploits the abstract and general model available in the RAMON platform.

The rest of the work contains a brief description of the overall RAMON architecture (Section 2), aimed at stating our original context for the radio resource management algorithms. Section 3 defines the general, platform independent radio resource assignment algorithm, by exploiting RLC, MAC and physical layer information (cross-layer design). Section 4 describes in detail the adaptation modules to fit the general algorithm onto two specific REs, UMTS and Bluetooth. Section 5 shows some performance results. Final remarks and further work are addressed in Section 6.

2 Functional Architecture

RAMON allows the deployment of control algorithms independent of the considered RE. A *Mobile Station* (MS) or a *Base Station* (BS), equipped with the RAMON software module, supports abstract functionalities common to all the REs. Such functionalities are obtained from the specific operations of each RE, by means of translation entities, as described below.

Figure 1 shows the RAMON functional architecture. It includes a *Common Control*

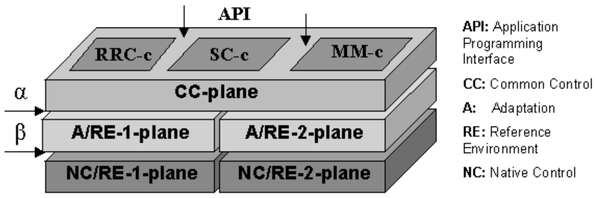

Fig. 1. Overall system architecture.

plane (CC-plane) and an *Adaptation plane* (A-plane). *Native Control* functionalities (NC-plane) for each RE (denoted as NC/RE-*i* for the *i*-th RE) are located below the A-plane. A service development RE-independent API (*Application Programming Interface*) is offered to the overlying *Application Layer*. CC-plane functions, which are defined independently of the underlying RE, are grouped in three functional sets, according to the classical model adopted for wireless communications: i) *Radio Resource Control* (RRC-c); ii) *Session Control* (SC-c); iii) *Mobility Management* (MM-c) [1];

The A-plane, which is divided into as many parts (A/RE-*i*) as the number of different REs involved, translates: 1) Primitives exchanged between the CC-plane and the NC-plane for each RE; 2) *Control Parameters* (CP) data passed from the *Native User plane* (NU-plane) and the CC-plane. Two different types of interfaces can be identified: i) the α interface, between the CC-plane and the A-plane; ii) the β interface, between the A-plane and the NC-plane of each RE. Figure 2 completes Figure 1. The CC-plane and the A-plane with the relevant α and β interfaces are depicted on the right. The NC-planes and NU-planes are shown on the left. Figure 2 clarifies how the SC-c, MM-c and RRC-c functionalities interact with the corresponding SC-*i*, MM-*i* and RRC-*i* (*i*=1,2) of the two REs through the translation performed by the A planes (A/RE-1, A/RE-2). The relations between the NC-planes and NU-planes are highlighted and particularly how the SC-*i*, MM-*i* and RRC-*i* functionalities are related with Physical (PHY-*i*), *Medium Access Control* (MAC-*i*) and *Radio Link Control* (RLC-*i*) functions is shown. Finally, Applications, TCP/UDP and IP layers, which are part of the NU-plane, directly interact with the CC-plane. The main idea is to dynamically adapt the whole stack to the perceived link quality by means of adequate control actions. Quality measurements are passed to the RAMON entity by means of CP data.

Cross-layer interaction requires control information sharing among all layers of the protocol stack in order to achieve dynamic resource management. It may be performed in two directions: i) *upward information sharing*: upper layers parameters may be configured to adapt to the variable RE characteristics; ii) *downward information sharing*: MAC, RLC and PHY layer parameters can adapt to the state of Transport and Application layer parameters.

[1] -c means in the common control plane.

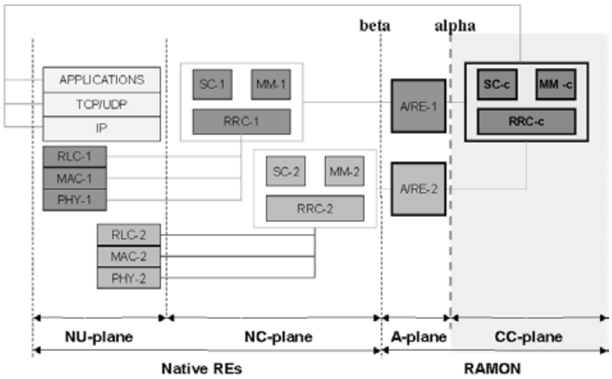

Fig. 2. Basic interactions among Control and User planes.

3 Common Control Plane and Reconfigurable Algorithm

This Section is dedicated to describe the general CHAOS scheme in the framework of the RAMON architecture.

CHAOS is part of the RRC functional set of the CC-plane of the RAMON architecture. Thus it is defined independently of the underlying REs. The scheme sets a framework for designing scheduling strategies for efficient resource sharing control in a scenario constituted by a single cell served by a BS, based on an “abstract” representation of i) physical channel state; ii) traffic condition for each MS involved in the communication process. This representation is expressed by means of abstract quantized parameters, *CHANNEL STATE* and *TRAFFIC CONDITION* respectively. The system state can be described by four vectors of length N (where N is the number of MSs in the scenario):

1. CHANNEL_STATE for each MS (downlink);
2. CHANNEL_STATE for each MS (uplink);
3. TRAFFIC_CONDITION for each MS (downlink);
4. TRAFFIC_CONDITION for each MS (uplink).

Figure 3 shows the basic information flow, i.e., how the CHAOS entity running in the CC-plane gathers information about system state. For every MS, in both directions, the RRC-*i* (relative to the *i*-th RE) entity in the NC-plane of the BS collects information that will be used for the evaluation of the abstract parameters by the A-plane. The RRC-*i* entity on the BS gets the BS-related information directly from the different layers of the NU-plane (i.e. the RLC-*i*, MAC-*i* and PHY-*i* layers). Information lying in the MS has to be exchanged between the two RRC peer entities (MS RRC-*i* and BS RRC-*i*) by means of the transport facilities offered by lower layers. As soon as new information describing the system state is acquired by the BS RRC-*i*, a *CP UPDATE indication primitive* (1), which is RE-dependent, is issued to the A-plane.

The mapping of these RE-dependent parameters into abstract parameters (CHANNEL STATE and TRAFFIC CONDITION), that can be used by the CHAOS algorithm,

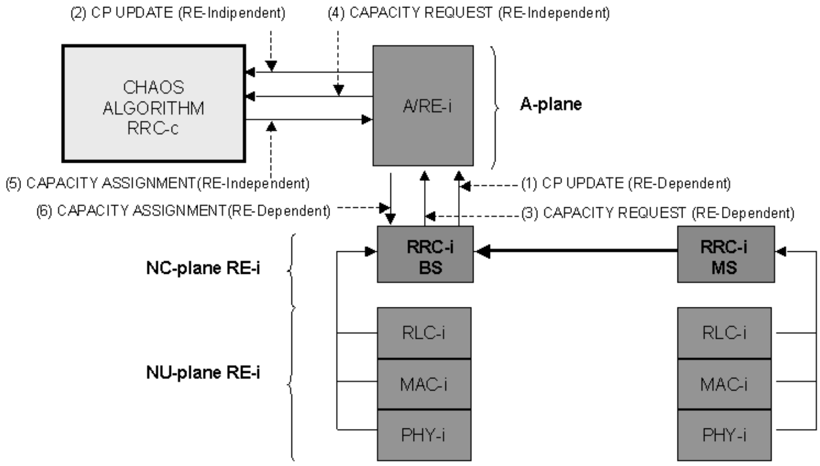

Fig. 3. Description of Information Flow from U-plane to CC-plane.

is up to the A-plane, which asynchronously communicates the abstract parameters to the CC-plane entity as soon as they are available, by means of an RE-independent *CP UPDATE primitive* (2).

The A/RE-*i* entity asynchronously issues RE-independent *CAPACITY ASSIGNMENT REQUEST primitives* (4) to the CHAOS entity in the CC-plane. These abstract requests are formulated by the A/RE-*i* plane on the basis of the RE-dependent explicit request coming from the RRC-*i* (RE-Dependent *CAPACITY REQUEST primitive* (3)) or on the basis of the CPs gathered when no explicit requests are issued by native system control mechanisms. These primitives contain the overall amount of capacity to be assigned. The CHAOS entity processes this request and returns an ordered vector of length N_T ($N_T \leq N$) whose element contains the MS *id* and the amount of capacity assigned to it through the RE-independent *CAPACITY ASSIGNMENT primitive* (5). The last step (6) consists in the A/RE-*i* entity translating the assignment into an RE-dependent *CAPACITY ASSIGNMENT primitive* command that can be issued to the BS NC-plane. This will result in the actual assignment with NC-plane specific mechanisms.

As a matter of example, the channel state could be a normalized measure of the distance between the target *Signal to Noise Ratio* (SNR) of the link and its actual SNR; or it could be derived from *Bit Error Rate* measurements. For instance, we could have three channel states (good, fair, bad). Analogously, traffic condition could be a normalized measure of the age of the packets or backlog.

We will finish this section by describing how the CHAOS entity uses system state information to assign capacity to the different MSs. As soon as *CAPACITY REQUEST primitives* arrive from the A/RE-*i* plane, this information is arranged in a matrix. In Figure 4 the CHAOS matrix is shown with three different scanning methods. The horizontal dimension represents the channel state, and thus requests with a different value of CHANNEL STATE occupy different positions in the matrix. The vertical dimension

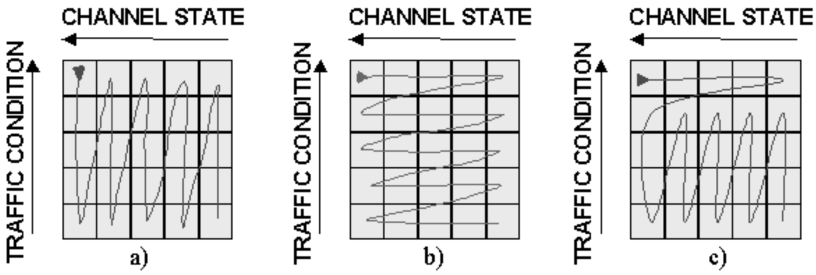

Fig. 4. The CHAOS matrix. Examples of matrix scanning methods

is associated with traffic condition, thus requests with different values of TRAFFIC CONDITION are put in different vertical positions in the matrix. What the CHAOS entity obtains is thus a two-dimensionally ordered description of the requests coming from different MSs. User requests are served in the order defined by a predefined rule (scanning method). Each different rule results in a different scheduling discipline. In Figure 4(a) the matrix is scanned by column by giving priority to traffic state conditions; in Figure 4(b) it is scanned by row by giving priority to channel state conditions; a combination of the first two methods is shown in Figure 4(c).

Thus, CHAOS can be better understood as a class of algorithms, each qualified by a different service

discipline, which is somehow computed on the basis of the matrix.

4 Adaptation Planes and Reference Environments

In this Section the RE-dependent A-planes for the UMTS and Bluetooth systems are described. Native mechanisms and procedures are introduced and some design choices are explained. After a brief description of the capacity allocation procedures of the two REs (Subsections 4.1 and 4.3), we illustrate how to adapt these mechanisms to uniform them to the CHAOS paradigm (Subsections 4.2 and 4.4)).

4.1 UMTS-TDD

In UMTS-TDD air interface the time axis is slotted into *Time Slots* (TSs) and 15 TSs are grouped into a $10\,ms$ frame. Within each time slot a set of variable spreading factor orthogonal codes are defined to spread QPSK modulated data symbols of up to 16 different users. The overall capacity of the radio interface is shared by common channels and data channels. In our study the *Random Access CHannel* (RACH) is allocated to the first TS of every frame and the *Forward Access CHannel* (FACH) and *Broadcast CHannel* (BCH) are allocated to the last one. We assume that transport blocks are 320 bit long, (including MAC/RLC overhead); a *Transmission Time Interval* (TTI) of $10\,ms$ and a convolutional coding with rate $1/2$ are used. This implies that four codes with spreading factor 16 are required to carry the coded transport block; the choice of this transport format implies that the minimum assignable capacity is $32\ kbit/s$ which will be referenced to in the sequel as *Resource Unit* (RU). From a modeling point of view, the entire transport resource of the radio interface (i.e. 60 RUs) is divided into common channels (8 RUs) and shared channels (52 RUs) for data transfer. Each slot can carry up to four data (plus associated signaling) transport blocks in each frame, if we require that the four RUs used by a transport block be all in the same time slot.

Different kind of logical channels are available for data transfer. The choice depends on the traffic type and on traffic volume. *Dedicated CHannels* (DCH) are well suited for *Real Time* services (RT), whereas common channels (RACH and FACH) or shared channels (*Uplink Shared CHannel* (USCH) and *Downlink Shared CHannel* (DSCH)) are suited for *Non Real Time* services (NRT). For mobile computing purposes (web-like and file transfer applications) the DSCH and the USCH are the most appropriate choices. These channels are allocated for a limited time and are released automatically. Figure 5 shows the data transfer procedure on USCH [16]. After scanning the RLC buffer, the MS sends a *Capacity Request* message to the UTRAN; this message contains a traffic measure (the RLC buffer length) and it is sent on the SHCCH (*SHared Control CHannel*), mapped on the RACH channel or on a USCH (if there is one already allocated to the MS). When the UTRAN RRC layer runs the *Scheduling Algorithm*, a resource allocation message (*Physical Shared Channel Allocation*) is sent to the MS on the SHCCH, which is mapped on the FACH or on the DSCH. At this point the MS can start data transfer. The allocation message allocates a number of RUs for a number t of frames (with $1 \leq t \leq t_{max}$). For this reason, it is not necessary to acknowledge the message. A similar procedure is employed for DSCH allocation.

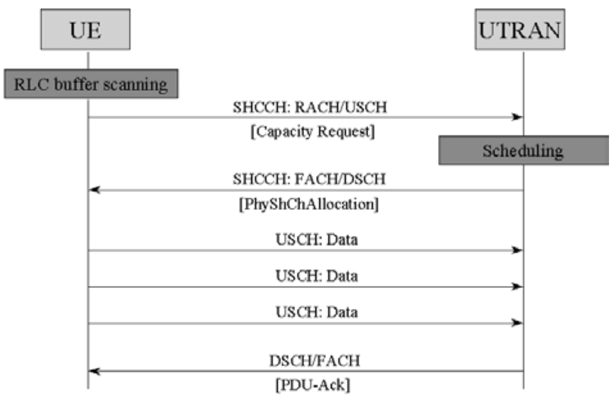

Fig. 5. USCH channel allocation procedure.

4.2 UMTS-TDD Adaptation Plane

In this Section the interactions of the A/UMTS-plane with the UMTS-RRC entity and with RRC-c entity are described. On the one hand the UMTS-TDD Adaptation plane reads informations about channel and traffic conditions of MSs from the RRC-UMTS layer, processes these informations and informs the RRC-c plane about these updates, on the other hand it requests RRC-c plane for the resource allocation, maps it in a UMTS native language and passes it to the RRC-UMTS. To perform all these operations the plane must be aware of the UMTS resource allocation procedures (exchanged messages, parameters, etc.) and must interact with the RRC-UMTS entity to get all needed informations and to replace the RRC allocation algorithm. There are two basic procedures which can be better explained with two examples.

- *CP-UPDATE*. When the RRC-UMTS layer receives a new allocation request from a MS (see Figure 3 (3)), or when it receives new packets directed to a MS, it sends a CP-UPDATE command to the RRC-c plane with the direction link (UL or DL), the mobile identity and the channel quality. The queue length indicated in the request is sent in a CAPACITY REQUEST (see Figure 3(4)). The RRC-c functional entity updates the matrix.
- *CAPACITY REQUEST*. Every 10 ms the A-plane sends a CAPACITY ASSIGNMENT REQUEST to the RRC-c entity indicating the unallocated capacity of the frame; the RRC-c scans the matrix and creates a vector with the MS identities, the capacity to allocate in one frame and the duration of the allocation. The A-plane translates it into the UMTS-TDD language (i.e. TSs, codes to assign and number of frames) and sends new allocations to the RRC-UMTS that will inform the involved MSs. As seen in the second example, the CAPACITY ASSIGNMENT REQUEST indication primitive is sent one time per frame and lets the RAMON module synchronize with the UMTS frame. The UPDATE messages, instead, are totally asynchronous and can be transmitted at every instant. From the RAMON point of view the whole UMTS-TDD system can be seen as shared capacity that varies in time but remains fixed in the frame.

4.3 Bluetooth MAC

In this paragraph the basic issues related to the MAC technique adopted in the Bluetooth (BT) technology are presented. Devices are organized into piconets, i.e., small networks which include up to 8 nodes. Different piconets share the same channel (2.4 GHz ISM band) by exploiting the FHSS (Frequency Hopping Spread Spectrum) technique, i.e., by following different hopping sequences.

In a piconet, one device has the role of master, all the other devices are slave. Time is slotted ($625 \mu s$), the master controls the traffic and takes care of the centralized access control. Only master-slave communications are allowed. Odd slots in the TDD slotted structure are reserved for master transmission. The slave that receives a packet from the master is implicitly polled for the following even slot. Thus, the master controls the medium access by sending data to different slaves in even slots and implicitly authorizing the slave to transmit starting from the following slot. If the master has no data to send, it can authorize slaves transmission with an explicit POLL packet. Three data packet sizes are allowed: one slot, three slot, and five slot length.

4.4 Bluetooth Adaptation Layer

While the RRC-UMTS provides much information that can be easily adapted for the CHAOS scheduling, Bluetooth lacks mechanisms to retrieve physical layer (both in the UL and DL directions) and traffic state informations (for the UL direction). A great effort has been done to define a suitable A-plane[2] that would make up for the lack of native system informations. Moreover, due to the particular polling mechanism adopted in Bluetooth, master-slave pairs have to be jointly represented in a CHAOS matrix, since a slot assigned in the downlink direction implies at least a slot being assigned in the uplink. For every Master-Slave pair, the master keeps memory of the average Packet Error Rate (PER) on the link, by mediating on the last N_{tx} transmissions. This information is passed to the A-Plane which classifies the slaves into different classes of CHANNEL STATE according to their PER values.

Traffic condition for each link can be expressed by the amount of data to be transmitted in the L2CAP queue [17]. However, although the master node exactly knows queue-length for any of its slaves in the downlink direction (DLQ_i), uplink queue state must be somehow signaled by the slaves. We propose, as in [17], to use the flow bit in the Baseband packet header, which was intended in the Bluetooth Specifications [5] to convey flow information at the L2CAP level. In a way similar to [17], we use this bit to estimate the queue length of the slaves for the uplink direction, (ULQ_i). The master-slave pair, in the scheduling process, will be then represented by its *Virtual Queue*, defined as $VQ_i = ULQ_i + DLQ_i$. This parameter is then quantized by the A-plane and passed as TRAFFIC_CONDITION parameter to the CHAOS RRC-c entity. Due to the particular polling mechanism adopted in Bluetooth, the resource to be shared among the different M-S couples can not be easily described as capacity. The scheduler only decides which M-S couple has the right to transmit on the next available slot, but the amount of capacity assigned to each couple strongly depends also on other factors, such as the SAR (Segmentation & Reassembly), since choosing different packet sizes leads to different capacities being assigned.

An additional variable is defined in the A/BT-plane for each M-S couple, namely *Virtual Time*, to adapt the CHAOS framework to the polling mechanism. The Virtual Time assigned to the i-th M-S couple at step t is defined as:

$$V_t^i = V_{t-1}^i + \frac{l_{t-1}}{r_t^i} \tag{1}$$

where r_t^i represents the abstract capacity assigned by CHAOS to the i-th couple at step t and l_{t-1} is the duration of the last transmission of the M-S couple (which ranges from $625 \cdot 2\mu s$ if a 1 slot packets is used in both directions to $625 \cdot 10\mu s$ if both transmitters use a 5 slot packet). Thus, when CHAOS assigns high values of capacity, the *Virtual Time* increases slowly after each transmission, where as when it assigns high values of

[2] The authors would like to thank M. D. Marini for his precious work in adapting CHAOS to Bluetooth and for the simulation results presented in Section 5.2.

capacity it increases in a faster way. After one couple's transmission, the scheduler gives the right to transmit to the M-S couple with the minimum value of Virtual Time. This results in a higher number of chances to transmit being given to M-S couples CHAOS has assigned high capacity.

CHAOS assigns capacity on the basis of functions that can be computed on the matrix representing the system state, which somehow represent the scheduling discipline chosen. For the simulations presented in the following paragraph, a 5x5 matrix has been used to describe the system state. CHANNEL_STATE (C_S) and TRAFFIC_CONDITION (T_C) assume discrete values ranging from 1 to 5. The capacity assigned to the i-th couple is then calculated as:

$$r_t^i = \frac{T_C^2}{C_S^3} \tag{2}$$

which results in high capacity being assigned to couples with low values of CHANNEL_STATE (good channel) and high values of TRAFFIC_CONDITION (high traffic). However, channel state is given a higher exponent because the main target is avoiding potentially power-wasting transmissions when the state of the channel is bad.

5 Results

In this Section results about improvements that can be achieved with the CHAOS strategies are shown for the UMTS and for Bluetooth. Results have been obtained using an UMTS module for the ns network simulator [18] and extending the *Bluehoc* [19] functionalities. As said before it is important to notice that CHAOS constitutes a large class of algorithms and by modifying the matrix scanning methods different service disciplines (with different targets) can be easily achieved. This can be simply obtained by different scanning methods.

5.1 UMTS

In the simulated scenario N mobiles in a single cell send data through a gateway to N wired nodes. The entire data traffic is in the uplink direction. In the UMTS frame 9 time-slots have been statically allocated to the uplink (1 for the RACH and 8 for the USCHs), and the remaining 6 have been allocated to the downlink (1 for the FACH and BCH and 5 for the DSCH); thus, the maximum attainable throughput in the uplink direction is $1024\ kbit/s$. The traffic is generated by FTP traffic sources (one FTP agent per node) on a TCP transport layer.

Three matrix scanning methods have been used in simulations; the first one, named $Random$, scans the matrix randomly; the second one, named $CHAOS_1$, scans the matrix by giving priority to the oldest requests (see Figure 4(b)), and the last one, $CHAOS_2$, gives priority to the requests from users with better channel quality (see Figure 4(a)). In Figure 6 we can see that throughput is maximized by giving priority to "best channel" requests. This throughput gain is due to more efficient use of the radio interface: the adoption of a channel adaptive scheduling decreases the RLC-PDU error

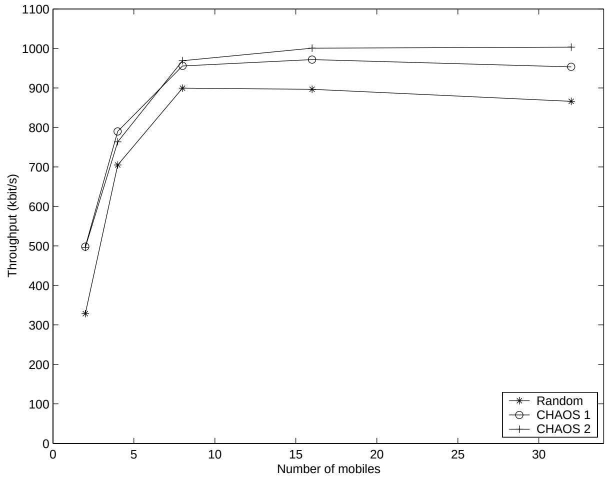

Fig. 6. FTP throughput vs. number of MSs varying matrix scanning methods.

Retransmissions	Algorithm	
	Random	**CHAOS**
0	95.676	99.888
1	2.765	0.1
2	0.98	0.011
3	0.363	0.00099
4	0.128	0.00013
5	0.05	0
≥ 6	0.068	0

Table 1. Percentual distribution of RLC PDU retransmissions

probability, leading to a better exploitation of the radio resource. This in turn reduces the need for packet retransmissions, as can be seen from Table 1. Figure 7 illustrates the gain in energy efficiency brought about by the use of a channel-adaptive packet scheduling algorithm. This result can be explained by observing that, with the CHAOS algorithm, mobiles tend to transmit more during the intervals in which they experience a high channel quality, when the transmitted power is reduced by the power control algorithm. Moreover, the decrease in the number of retransmissions also contributes to the increase in efficiency.

5.2 Bluetooth

The simulation scenario involves a piconet with one master and two slaves. Two different CBR connections are supported in the downlink direction (from master to slave), and transport and network layer protocols are respectively UDP and IP. The channel is modeled as a two-state Markov Chain: in the BAD state the *Packet Error Rate* PER

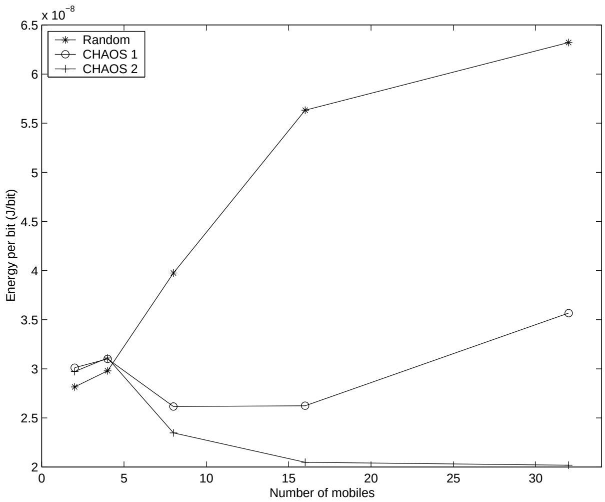

Fig. 7. Energy per bit vs. number of MSs varying matrix scanning methods.

is very high (90%) where as in the GOOD state no errors occur on the channel. Residence time in each state is exponentially distributed with mean residence time equal to 5 seconds in the BAD state and 20 seconds in the GOOD state. This model tries to account for interference from co-located 802.11b devices. Figure 8 shows the throughput of the piconet with respect to the load offered to the piconet, normalized to the capacity of the piconet itself. CHAOS is shown to reach a better throughput with respect to a Deficit Round Robin (DDR) scheduler, especially as the load increases. Figure 9 shows the number of information bits received with respect to information bits transmitted obtained by varying the offered load to the piconet. This value can be interpreted as a measure of power efficiency since for low power devices (CLASS 2) no power control is used. It can be observed that while CHAOS constantly outperforms the DRR scheduler, higher values of power efficiency are obtained when offered load increases. This mainly happens because when much free capacity is available (which happens for lower values of offered load) useless retransmissions occur, which fail because the channel is in the BAD state.

6 Final Remarks And Further Work

In this paper a scheme for developing reconfigurable scheduling disciplines was presented. A modular architectural paradigm was introduced, which allows the deployment of general algorithms based on simple modules that abstract functionalities common to the wireless systems in use today. We exploited this architecture to develop scheduling algorithms adaptive to both traffic and channel quality; these algorithms are then applied to two different Reference Environments (Bluetooth and UMTS), and they are shown to achieve an efficient utilization of the radio interface. This occurs because the

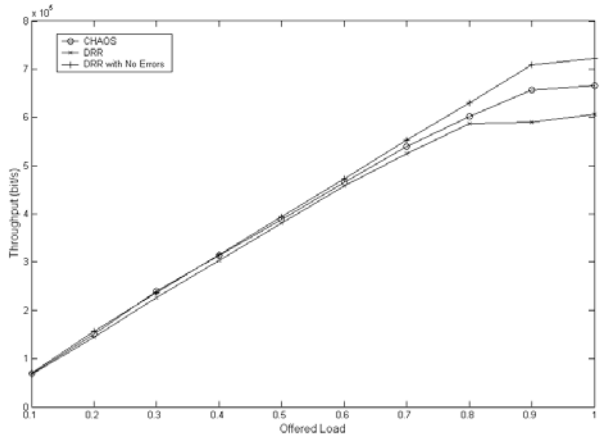

Fig. 8. Throughput vs Offered Load with different scheduling disciplines.

adoption of a cross layered architecture enhances native system functionalities. The adaptation plane algorithms for the two reference environments have been specified in detail. Results for each RE show how scheduling algorithms manage to reap the best from different REs.

References

1. Roveri A., Chiasserini C.F., Femminella M., Melodia T., Morabito G., Rossi M., Tinnirello I.: "The RAMON module: architecture framework and performance results," published in these Proceedings.
2. Haas Z.J.: "Design Methodologies for Adaptive and Multimedia Networks," IEEE Communications Magazine., Vol. 39, no. 11, November 2001.
3. Rappaport T.S., Annamalai A., Buehrer R.M., Tranter W.H.: "Wireless Communications: Past Events and a future Perspective," IEEE Commun. Mag., Vol 39, no. 11, May 2002.
4. Haardt M., Klein A., Kohen R., Oestreich S., Purat M., Sommer V., Ulrich T.: "The TDCDMA Based UTRA TDD Mode," IEEE Journal on Selected Areas in Communications, vol. 18, no. 8, August 2000.
5. Bluetooth Special Interest Group: "Specifications of the Bluetooth System, Volume 1, Core. Version 1.1, February 22 2001.
6. Bharghavan V., Lu S., Nandagopal T.: "Fair Queuing in Wireless Networks: Issues and Approaches," IEEE Personal Communications, vol. 6, no. 1, February 1999.
7. Lu S., Bharghavan V., Srikant R.: "Fair Scheduling in Wireless Packet Networks," IEEE/ACM Trans. on Networking, vol. 7, no. 4, August 1999.
8. Nandagopal T., Kim T. E., Gao X., Bharghavan V.: "Achieving MAC Layer Fairness in Wireless Packet Networks," Proc. of ACM MOBICOM 2000, August 6-11 2000, Boston, Massachussets.
9. Liu Z., Karol M.J., El Zarki M., Eng K.Y.: "Channel access and interference issues in multicode DS-CDMA wireless packet (ATM) networks," ACM Wireless Networks, 2, 1996.

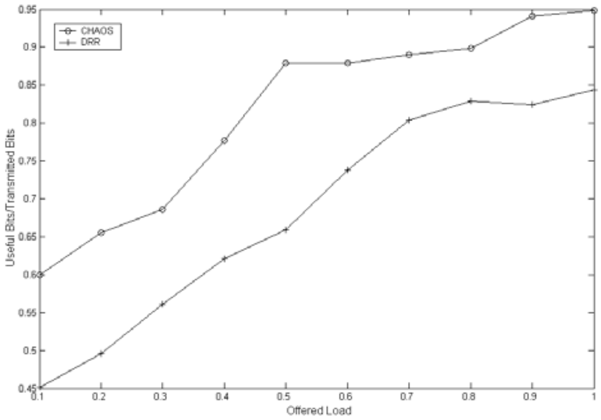

Fig. 9. Received Bits/Transmitted Bits vs Offered Load with different scheduling disciplines.

10. Alonso L., Agust R., Sallent O.: “A Near-Optimum MAC Protocol Based on the Distributed Queueing Random Access Protocol (DQRAP) for a CDMA Mobile Communication System,” IEEE Journal on Selected Areas in Communications, vol. 18, no. 9, September 2000.
11. Elaoud M., Ramanathan P.: “Adaptive Allocation of CDMA Resources for Network-level QoS Assurances,” Proc. of ACM MOBICOM 2000, August 6-11 2000, Boston, Massachussets.
12. Lee S. J., Lee H. W., Sung D. K.: “Capacity of Single-Code and Multicode DS-CDMA System Accommodating Multiclass Services,” IEEE Transactions on Vehicular Technology, vol. 48, no. 7, March 1999.
13. Bhagwat P., Bhattacharya P., Krishna A., Tripathi S.K.: “Enhancing throughput over wireless LANs using channel state dependent packet scheduling,” Proc. of IEEE INFOCOM 96, April 2-6 1996, San Francisco, California.
14. Fragouli C., Sivaraman V., Srivastava M.B.: “Controlled multimedia wireless link sharing via enhanced class-based queuing with channel state dependent packet scheduling,” Proc. INFOCOM 98, April 1998, San Francisco, California.
15. Baiocchi A., Cuomo F., Martello C.: “Optimizing the radio resource utilization of multiaccess systems with a traffic-transmission quality adaptive packet scheduling,” Computer Networks (Elsevier), Vol. 38, n. 2, February 2002.
16. 3GPP Technical Specification TS 25.331, V3.8.0 Release 99, “RRC Protocol Specification,” September 2001.
17. Das A., Ghose A., Razdan A., Saran H., and Shorey R.: “Enhancing Performance of Asynchronous Data Traffic over the Bluetooth Wireless Ad-hoc Network,”IEEE INFOCOM 2001, Alaska, USA, April 2001.
18. ns - Network Simulator, UC Berkeley. Internet Site: http://www.isi.edu/nsnam/ns.
19. Bluehoc Simulator Site: http://www-124.ibm.com/developerworks/opensource/bluehoc/.